**צוותי הרפואה בצה"ל**

שי דופלט 208737981

גלעד עבודי 208503060

**שלב א:**

**תיאור הארגון**

בפרויקט זה נבנה בסיס נתונים לצוותי הרפואה הגדודיים בצה"ל.

צוות רפואה גדוד נקרא תאג"ד (=תחנת איסוף גדודית) והוא מורכב ממספר בעלי תפקידים שלכל אחד ישנו תחום אחריות מסויים והם פועלים כצוות.

ישנו מפקד אחד, אחראי ציוד אחד, ובשאר התפקידים – רופא, פראמדיק, חובש ונהג יכולים להיות מספר אנשים בכל תפקיד.

מפקד – commander – אחראי על ניהול הצוות ומוודא שכל אחד מבצע את תפקידו בצורה המיטבית. מוגדר על ידי מספר זהות (comID) דרגה (rank) ומספר יחידה((numofunitcom.

אחראי ציוד – equipment manager - אחראי על ניהול מלאי הציוד ואספקתו לאנשי הרפואה בתאג"ד. מוגדר על ידי מספר זהות (equID), תחום התמחות(expertise) ומספר יחידה((numofuniteqi.

רופא – doctor - אחראי על הטיפול הרפואי בשגרה.

מוגדר על ידי מספר זהות(docID) ,ותק(seniority), מספר רשיון(license\_number) ומספר יחידה((numofunitdoc.

פראמדיק – paramedic - אחראי על הטיפול הרפואי בחירום.

מוגדר על ידי מספר זהות(parID), ותק(seniority) , מספר רשיון(license\_number) ומספר יחידה((numofunitpar.

חובש – medic - עוזר לפראמדיק בזמן הטיפול בחירום.

מוגדר על ידי מספר זהות(medID), ותק(seniority) ומספר יחידה((numofunitmed.

נהג – driver - אחראי על נהיגה באמבולנס להגעה ממקום למקום.

מוגדר על ידי מספר זהות(drID), מספר רשיון(license\_number) ומספר יחידה((numofunitdri.

בכולם מתקיים NF3

**הטבלאות המתקבלות:**

Commander (comID , rank, numofunitcom)

Equipment manager (equID, expertise, numofuniteqi)

Doctor (docID , seniority, license\_number, numofunitdoc )

Paramedic (parID, seniority ,license\_number, numofunitpar)

Medic (medID, seniority, numofunitmed)

Driver (drID, license\_number, numofunitdri)

תמונה שמכילה תרשים, שרטוט, קו, ציור

התיאור נוצר באופן אוטומטי

תמונה שמכילה תרשים, קו, מקביל, שרטוט טכני

התיאור נוצר באופן אוטומטי

**הגדרת הטבלאות בSQL:**

CREATE TABLE commander

(

comID INT NOT NULL,

rank VARCHAR(15) NOT NULL,

numofunitcom INT NOT NULL,

PRIMARY KEY (comID)

);

CREATE TABLE driver

(

license\_number INT NOT NULL,

drID INT NOT NULL,

numofunitdri INT NOT NULL,

comID,

PRIMARY KEY (drID),

FOREIGN KEY (comID) REFERENCES commander(comID)

);

CREATE TABLE equipment\_manager

(

equID INT NOT NULL,

expertise VARCHAR(15) NOT NULL,

numofuniteqi INT NOT NULL,

comID,

PRIMARY KEY (equID),

FOREIGN KEY (comID) REFERENCES commander(comID)

);

CREATE TABLE medic

(

medID INT NOT NULL,

seniority INT NOT NULL,

numofunitmed INT NOT NULL,

comID,

equID,

PRIMARY KEY (medID),

FOREIGN KEY (comID) REFERENCES commander(comID),

FOREIGN KEY (equID) REFERENCES equipment\_manager(equID)

);

CREATE TABLE doctor

(

docID INT NOT NULL,

license\_number INT NOT NULL,

seniority INT NOT NULL,

numofunitdoc INT NOT NULL,

comID,

equID,

PRIMARY KEY (docID),

FOREIGN KEY (comID) REFERENCES commander(comID),

FOREIGN KEY (equID) REFERENCES equipment\_manager(equID)

);

CREATE TABLE paramedic

(

parID INT NOT NULL,

license\_number INT NOT NULL,

seniority INT NOT NULL,

numofunitpar INT NOT NULL,

comID,

equID,

PRIMARY KEY (parID),

FOREIGN KEY (comID) REFERENCES commander(comID),

FOREIGN KEY (equID) REFERENCES equipment\_manager(equID)

);

**באמצעות הפקודה DESC:**

commander

+-------+----------+--------------+

| Name | Type | Nullable |

+-------+----------+--------------+

| COMID | INT | No |

| RANK | VARCHAR | No |

| NUMOFUNITCOM | INT | No |

+-------+----------+--------------+

equipment\_manager

+-------+-----------+--------------+

| Name | Type | Nullable |

+-------+-----------+--------------+

| EQUID | INT | No |

| EXPERTISE | VARCHAR | No |

| NUMOFUNITEQI | INT | No |

| COMID | INT | No |

+-------+-----------+--------------+

doctor

+---------------+-----------+--------------+

| Name | Type | Nullable |

+---------------+-----------+--------------+

| DOCID | INT | No |

| LICENSE\_NUMBER| INT | No |

| SENIORITY | INT | No |

| NUMOFUNITDOC | INT | No |

| COMID | INT | No |

| EVEID | INT | No |

+---------------+-----------+--------------+

paramedic

+---------------+-----------+--------------+

| Name | Type | Nullable |

+---------------+-----------+--------------+

| PARID | INT | No |

| LICENSE\_NUMBER| INT | No |

| SENIORITY | INT | No |

| NUMOFUNITPAR | INT | No |

| COMID | INT | No |

| EVEID | INT | No |

| MEDID | INT | No |

+---------------+-----------+--------------+

medic

+-------------+-----------+--------------+

| Name | Type | Nullable |

+-------------+-----------+--------------+

| MEDID | INT | No |

| SENIORITY | INT | No |

| NUMOFUNITMED| INT | No |

| COMID | INT | No |

| EVEID | INT | No |

+-------------+-----------+--------------+

driver

+---------------+-----------+--------------+

| Name | Type | Nullable |

+---------------+-----------+--------------+

| LICENSE\_NUMBER| INT | No |

| DRID | INT | No |

| NUMOFUNITDRI | INT | No |

| COMID | INT | No |

+---------------+-----------+--------------+

**יצירת נתונים לטבלאות והכנסתם לבסיס הנתונים:**

**מפקד:**

תמונה שמכילה צילום מסך, טקסט, קו

התיאור נוצר באופן אוטומטי

תמונה שמכילה טקסט, צילום מסך, גופן

התיאור נוצר באופן אוטומטי

**אחראי ציוד:**

תמונה שמכילה צילום מסך, טקסט, גופן, קו

התיאור נוצר באופן אוטומטי

תמונה שמכילה טקסט, צילום מסך, גופן

התיאור נוצר באופן אוטומטי

**רופא:**

תמונה שמכילה צילום מסך, טקסט, גופן, קו

התיאור נוצר באופן אוטומטי

תמונה שמכילה טקסט, צילום מסך, גופן

התיאור נוצר באופן אוטומטי

**פראמדיק:**

תמונה שמכילה צילום מסך, טקסט, גופן

התיאור נוצר באופן אוטומטי

תמונה שמכילה טקסט, צילום מסך, סגול, לבנדר

התיאור נוצר באופן אוטומטי

**חובש:**

תמונה שמכילה צילום מסך, טקסט, קו, גופן

התיאור נוצר באופן אוטומטי

תמונה שמכילה טקסט, צילום מסך, צבעוני

התיאור נוצר באופן אוטומטי

**נהג:**

תמונה שמכילה צילום מסך, טקסט, גופן, קו

התיאור נוצר באופן אוטומטי

תמונה שמכילה טקסט, צילום מסך, גופן

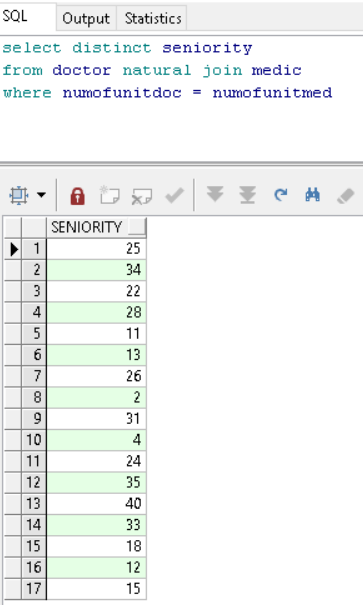
התיאור נוצר באופן אוטומטי

**שלב 2:**

שאילתות ללא פרמטרים:

1.

שנות הוותק ששוות בין רופא וחובש באותו הצוות:



2.

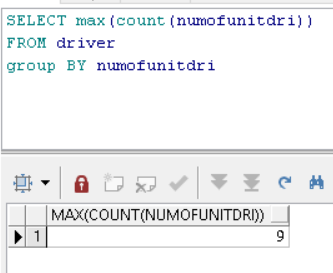
כמה מפקדים יש מכל דרגה – סגן,סרן ורב סרן?

תמונה שמכילה טקסט, צילום מסך, גופן, מספר

התיאור נוצר באופן אוטומטי

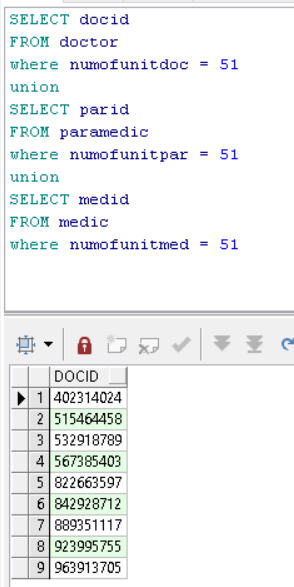
3.

כמה הכי הרבה נהגים יש באותה יחידה?



4.

מספרי הזהות של הרופאים, הפראמדיקים והחובשים ביחידה 51:



עדכון 1:

שינוי הוותק של החובשים בעלי 38 שנות נסיון ל 39 שנות נסיון.

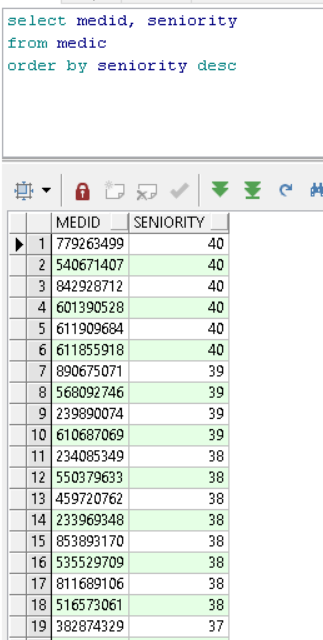
הפקודה:

UPDATE medic

SET seniority = 39

WHERE seniority = 38

צילום מסך לפני ואחרי:

 תמונה שמכילה טקסט, צילום מסך, מספר, גופן

התיאור נוצר באופן אוטומטי

עדכון 2:

שינוי דרגת המפקד עם מספר זהות 693509106 מדרגת סרן לדרגת רב סרן.

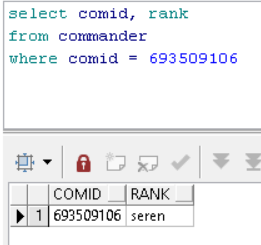
הפקודה:

UPDATE commander

SET rank = 'ravseren'

WHERE comid = 'seren'

צילום מסך לפני ואחרי:

מחיקה 1:

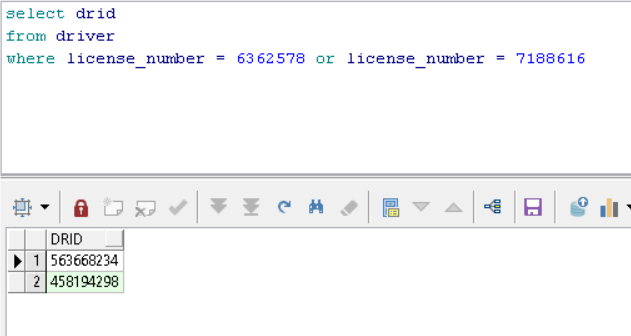
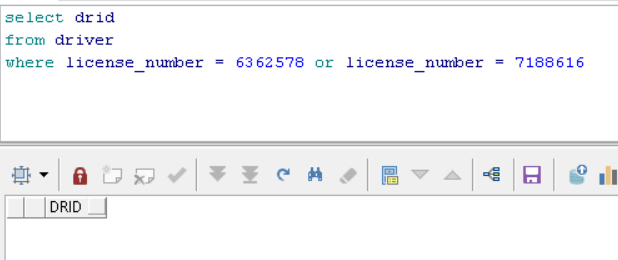
מחיקת הנהגים עם מספרי הרישיון – 6362578 ו- 7188616.

הפקודה:

Delete from driver

Where license\_number = 6362578 or license\_number = 7188616

צילום מסך לפני ואחרי:

מחיקה 2:

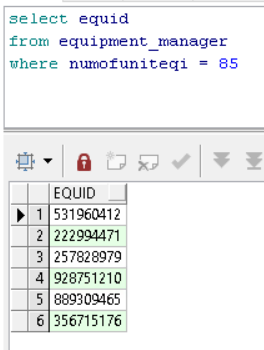
מחיקת כל אחראי הציוד ביחידה 87.

הפקודה:

Delete from equipment\_manager

Where numofuniteqi = 87

צילום מסך לפני ואחרי:

 תמונה שמכילה טקסט, צילום מסך, גופן, מספר

התיאור נוצר באופן אוטומטי

שאילתות עם פרמטרים:

1.

השאילתא מקבלת כקלט מספר זהות של מפקד ומחזירה את כל הפרטים שלו

תמונה שמכילה טקסט, גופן, קו, צילום מסך

התיאור נוצר באופן אוטומטי

תמונה שמכילה טקסט, צילום מסך, גופן, קו

התיאור נוצר באופן אוטומטי

2.

השאילתא מקבלת כקלט סוג התמחות של אחראי ציוד ומחזירה את הפרטים של כל אחראי הציוד באותו סוג התמחות

תמונה שמכילה טקסט, צילום מסך, גופן, קו

התיאור נוצר באופן אוטומטי

תמונה שמכילה טקסט, צילום מסך, גופן, מספר

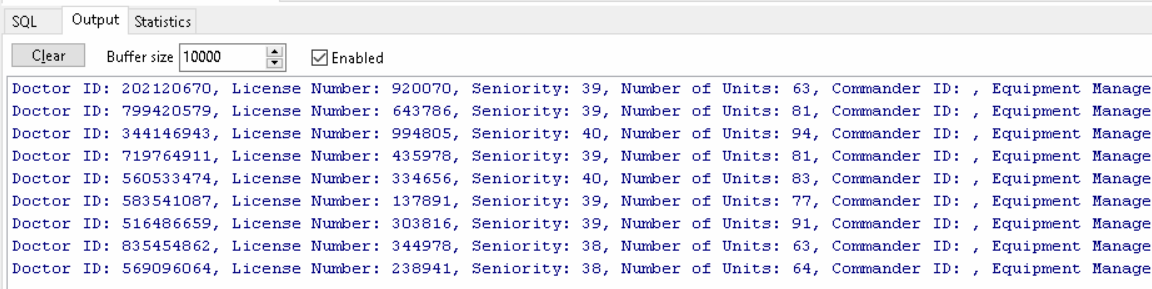
התיאור נוצר באופן אוטומטי

3.

השאילתא מקבלת כקלט מספר שנות וותק ומחזירה את הפרטים של כל הרופאים שמעל וותק זה

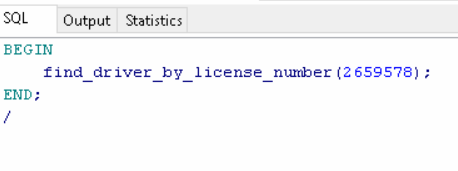
תמונה שמכילה טקסט, גופן, קו, מספר

התיאור נוצר באופן אוטומטי



4.

השאילתא מקבלת מספר רשיון של נהג ומחזירה את כל הפרטים שלו



תמונה שמכילה טקסט, צילום מסך, גופן, קו

התיאור נוצר באופן אוטומטי

אילוצים:

מספרי היחידות של מפקד יהיו תמיד בין 50 ל100:

ALTER TABLE commander

ADD CONSTRAINT check\_numofunitcom CHECK (numofunitcom BETWEEN 50 AND 100);

מספר רשיון של נהג לא יכול לחזור על עצמו יותר מפעם אחת:

ALTER TABLE driver

ADD CONSTRAINT unique\_license\_number UNIQUE (license\_number);

וותק של רופא, פראמדיק וחובש לא יהיה מעל 41:

ALTER TABLE doctor

ADD CONSTRAINT check\_doctor\_seniority CHECK (seniority <= 41);

ALTER TABLE paramedic

ADD CONSTRAINT check\_paramedic\_seniority CHECK (seniority <= 41);

ALTER TABLE medic

ADD CONSTRAINT check\_medic\_seniority CHECK (seniority <= 41);

**שלב ג:**

**1:**

**פונקציה המחשבת את מספר החיילים הכולל בצוותי הרפואה:** count\_entities

CREATE FUNCTION count\_entities RETURN INT IS

total\_count INT := 0;

temp\_count INT;

BEGIN

SELECT COUNT(\*) INTO temp\_count FROM commander;

total\_count := total\_count + temp\_count;

SELECT COUNT(\*) INTO temp\_count FROM driver;

total\_count := total\_count + temp\_count;

SELECT COUNT(\*) INTO temp\_count FROM equipment\_manager;

total\_count := total\_count + temp\_count;

SELECT COUNT(\*) INTO temp\_count FROM medic;

total\_count := total\_count + temp\_count;

SELECT COUNT(\*) INTO temp\_count FROM doctor;

total\_count := total\_count + temp\_count;

SELECT COUNT(\*) INTO temp\_count FROM paramedic;

total\_count := total\_count + temp\_count;

RETURN total\_count;

END count\_entities;

**פרוצדורה המעדכנת את הוותק של כל הפראמדיקים בשנה כלומר SENIORITY גדל ב1 ואם יש פראמדיק שהגיע ל41 שנות וותק היא מוחקת אותו מהמאגר:** update\_paramedics

CREATE OR REPLACE PROCEDURE update\_paramedics IS

BEGIN

UPDATE paramedic

SET seniority = seniority + 1;

DELETE FROM paramedic

WHERE seniority = 41;

END update\_paramedics;

**תכנית גלובלית הקוראת לפונקציה והפרוצדורה שעשינו:**

**היא בודקת את מספר החיילים הכולל בצוותי הרפואה, אחר כך מעדכנת את שנות הוותק של הפראמדיקים ומוחקת את מי שהגיע ל41 שנות וותק ובסוף מחזירה שוב את מספר החיילים הכולל בצוותי הרפואה לאחר העדכון.**

SELECT count\_entities() AS total\_entities\_before\_update FROM dual;

CALL update\_paramedics();

SELECT count\_entities() AS total\_entities\_after\_update FROM dual;

דוגמת הרצה:

בהתחלה הפונקציה מחזירה את מספר החיילים בצוותי הרפואה לפני העדכון:

תמונה שמכילה טקסט, צילום מסך, תוכנה, גופן

התיאור נוצר באופן אוטומטי

לאחר מכן הפרוצדורה השניה מעדכנת את הוותק של הפראמדיקים בשנה (+1) ואז מוחקת מהמאגר את מי שמעל 40 שנות וותק. בסוף הפונקציה מדפיסה את מספר החיילים לאחר העדכון, ורואים שהמספר ירד ב4 כלומר היו 4 פראמדיקים שהגיעו ל41 שנות וותק ונמחקו מהמאגר.

תמונה שמכילה טקסט, צילום מסך, תוכנה, גופן

התיאור נוצר באופן אוטומטי

**2:**

**פונקציה המחזירה את מספר הצוות עם מספר החיילים הגדול ביותר:** get\_unit\_with\_most\_staff

CREATE FUNCTION get\_unit\_with\_most\_staff RETURN INT IS

max\_unit INT;

BEGIN

SELECT numofunitcom

INTO max\_unit

FROM (

SELECT numofunitcom, COUNT(\*) AS staff\_count

FROM (

SELECT numofunitcom FROM commander

UNION ALL

SELECT numofunitdri AS numofunitcom FROM driver

UNION ALL

SELECT numofuniteqi AS numofunitcom FROM equipment\_manager

UNION ALL

SELECT numofunitmed AS numofunitcom FROM medic

UNION ALL

SELECT numofunitdoc AS numofunitcom FROM doctor

UNION ALL

SELECT numofunitpar AS numofunitcom FROM paramedic

)

GROUP BY numofunitcom

ORDER BY staff\_count DESC

)

WHERE ROWNUM = 1;

RETURN max\_unit;

END;

/

**פרוצדורה המקבלת את מספר הצוות הגדול ביותר בעזרת הפונקציה הקודמת ומעלה כל אחד מהמפקדים שבו בדרגה (מקסימום רב-סרן):** update\_commanders\_rank

CREATE PROCEDURE update\_commanders\_rank AS

BEGIN

FOR commander\_rec IN (SELECT comID, rank FROM commander WHERE numofunitcom = get\_unit\_with\_most\_staff()) LOOP

IF commander\_rec.rank = 'segen' THEN

UPDATE commander SET rank = 'seren' WHERE comID = commander\_rec.comID;

ELSIF commander\_rec.rank = 'seren' THEN

UPDATE commander SET rank = 'ravseren' WHERE comID = commander\_rec.comID;

END IF;

END LOOP;

END;

/

**תכנית גלובלית הקוראת לפונקציה והפרוצדורה שעשינו:**

**היא מוצאת את מספר הצוות עם מספר החיילים הגדול ביותר ולאחר מכן מבצעת את הפרוצדורה לעדכון דרגות המפקדים בצוות זה.**

DECLARE

most\_staff\_unit INT;

BEGIN

most\_staff\_unit := get\_unit\_with\_most\_staff();

update\_commanders\_rank();

END;

/

דוגמת הרצה:

הפעלת הפונקציה למציאת הצוות עם מספר אנשי הצוות הגדול ביותר:

תמונה שמכילה טקסט, צילום מסך, קו, גופן

התיאור נוצר באופן אוטומטי

נראה את רשימת ודרגות המפקדים בצוות זה על ידי הרצה שאילתה לפני ואחרי הרצת התכנית כדי שנוכל לראות את השינוי.

נריץ את התכנית הגלובלית המריצה את הפונקציה ואז את הפרוצדורה ונראה שבאמת בצוות המדובר דרגות המפקדים השתנו.

תמונה שמכילה טקסט, צילום מסך, גופן, קו

התיאור נוצר באופן אוטומטי

לפני: אחרי:

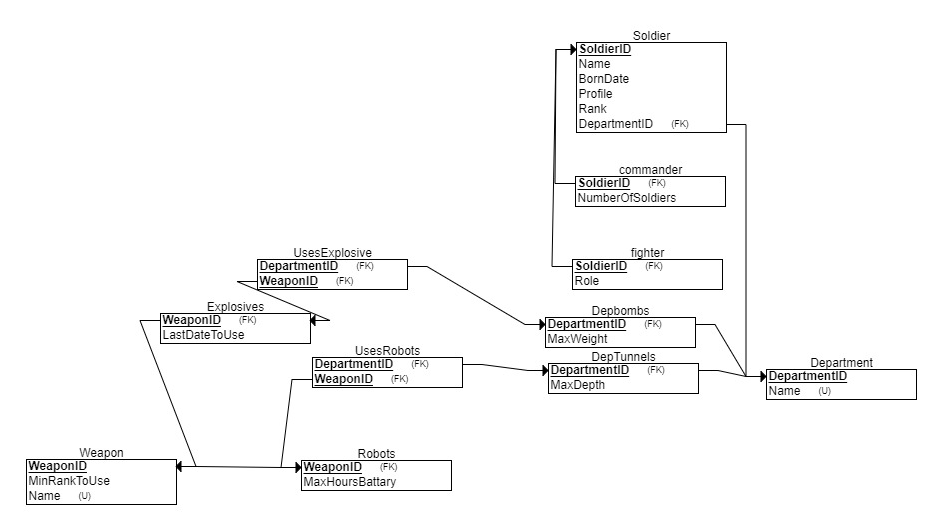
תמונה שמכילה טקסט, צילום מסך, גופן, תצוגה

התיאור נוצר באופן אוטומטי תמונה שמכילה טקסט, צילום מסך, תוכנה, גופן

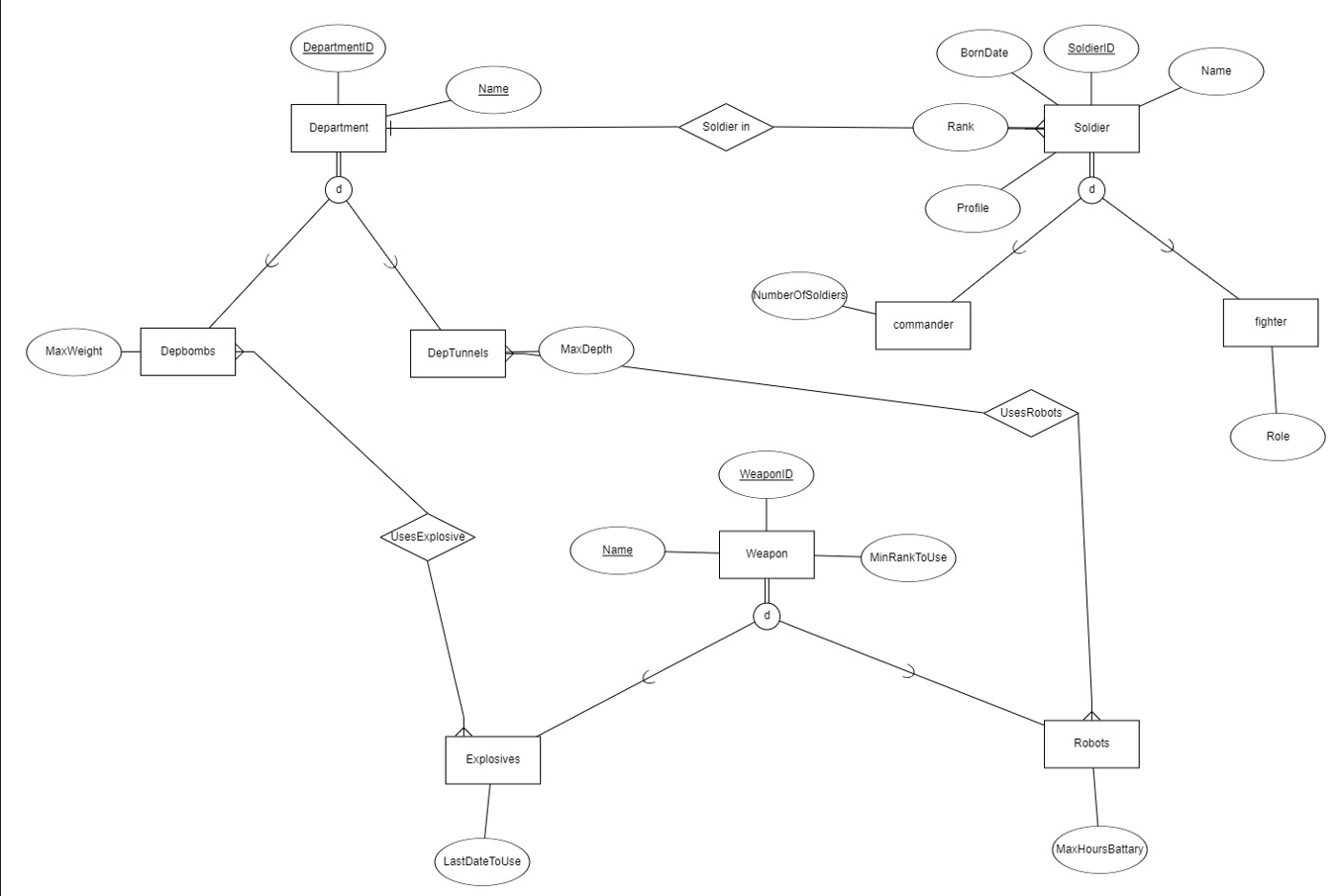
התיאור נוצר באופן אוטומטי

**שלב 4:**

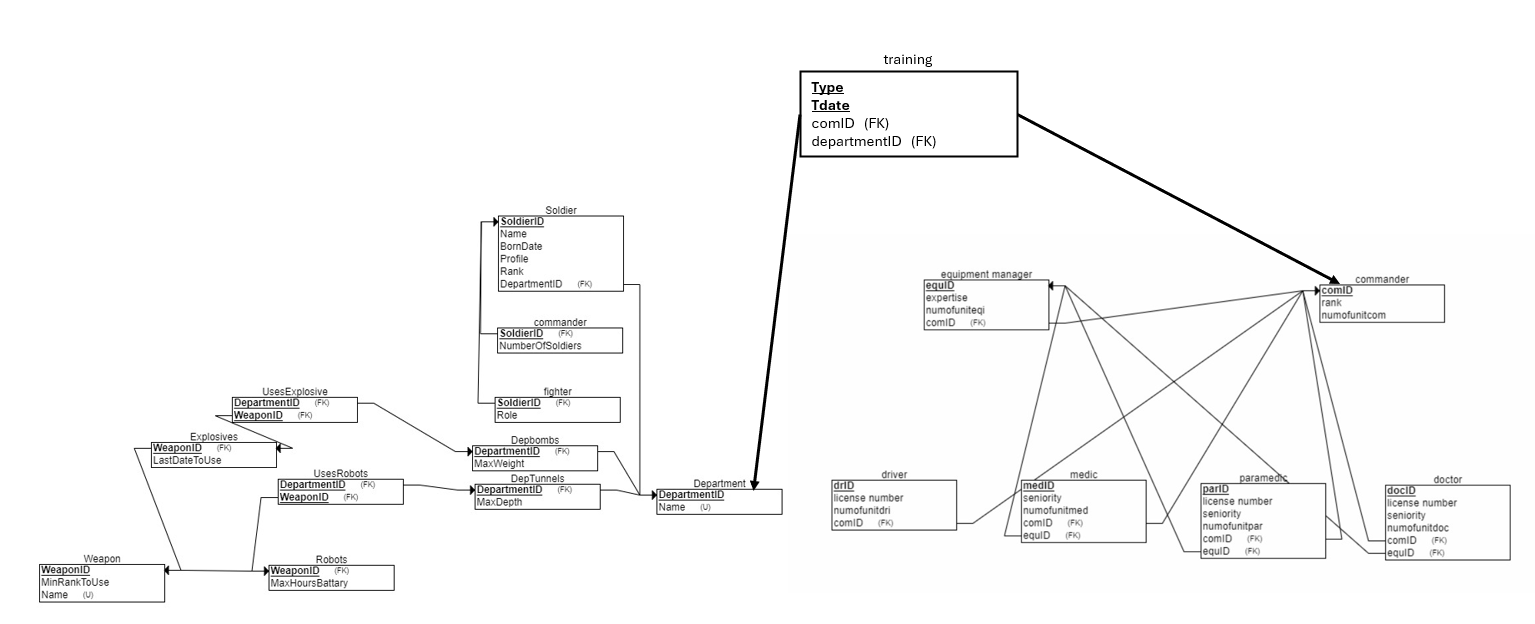
תרשים DSD של הפרויקט של חיל ההנדסה:



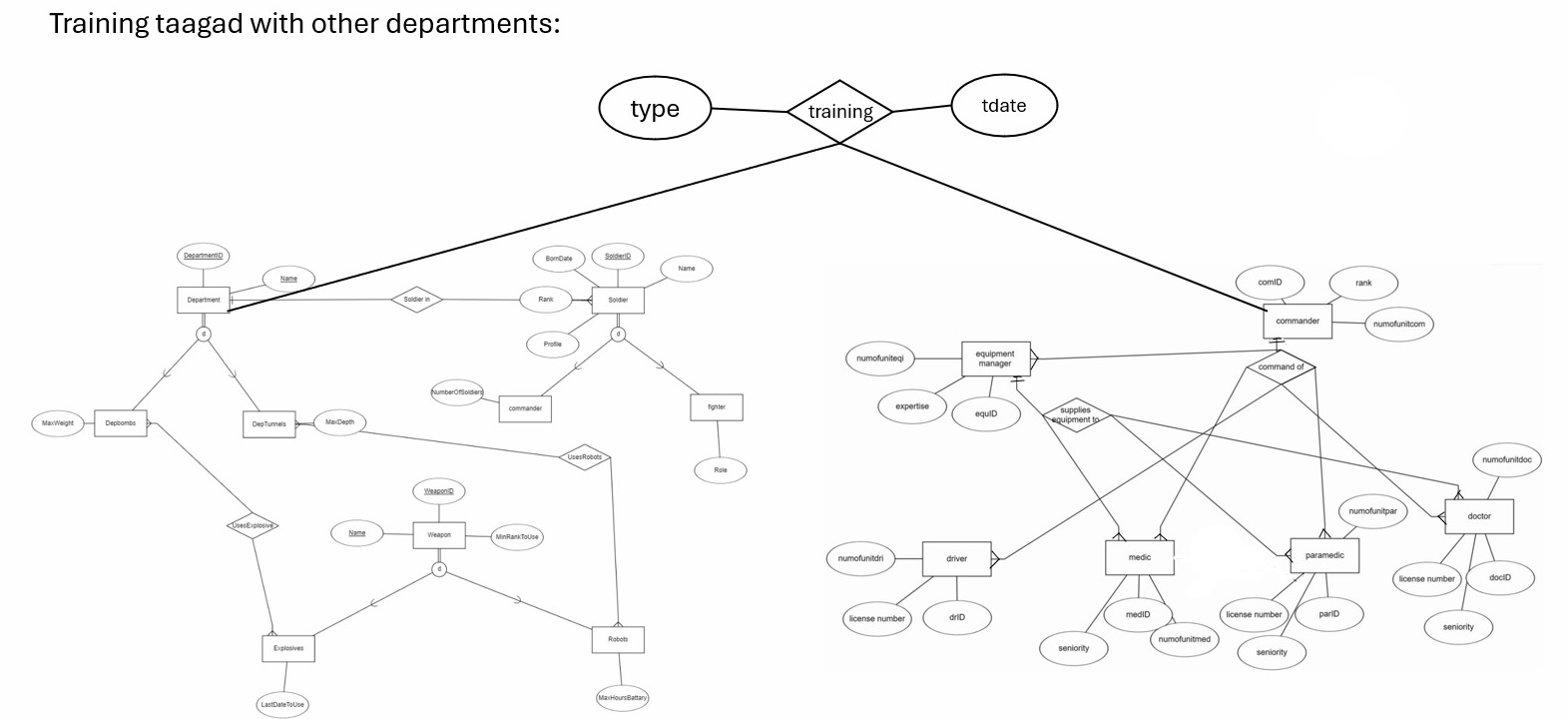
תרשים ERD של הפרויקט של חיל ההנדסה:



תרשים DSD של הפרויקט המשותף של התאג"ד וחיל ההנדסה:



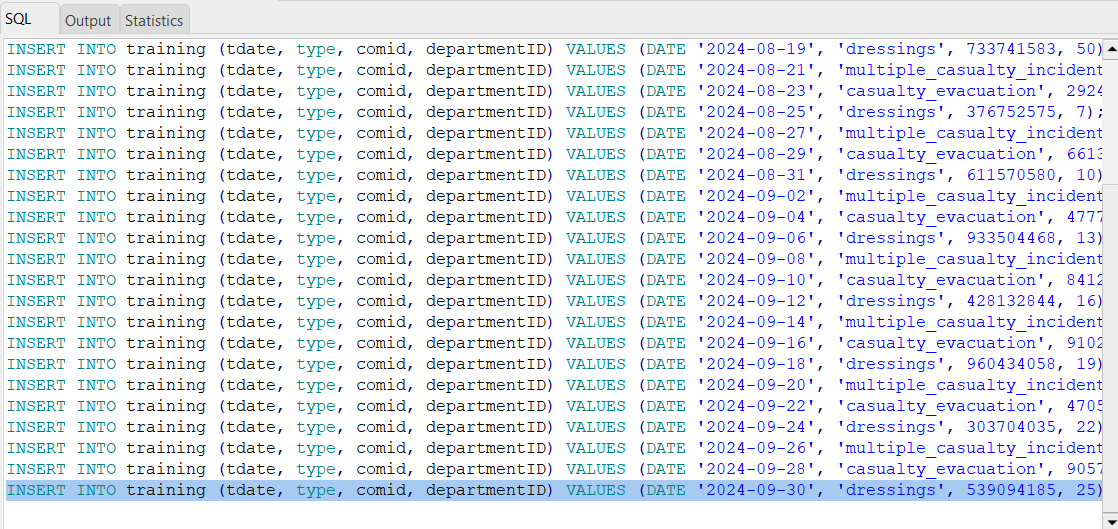
תרשים ERD של הפרויקט המשותף של התאג"ד וחיל ההנדסה:



יצירת טבלה המייצגת אימון של צוותי תאגד עם מחלקות חיל ההנדסה. האימון כולל מחלקת\מחלקות הנדסה ומפקד\מפקדי תאגד ויש לו סוג אימון ותאריך.

תמונה שמכילה טקסט, צילום מסך, גופן, תצוגה

התיאור נוצר באופן אוטומטי



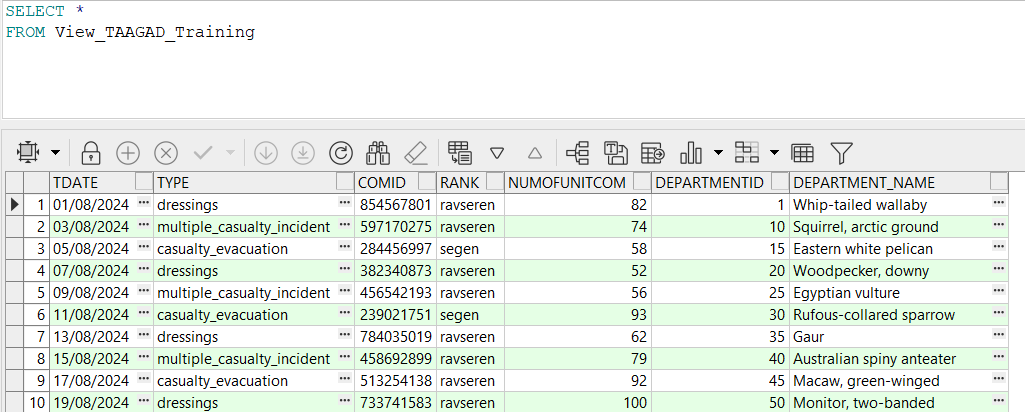
מבטים:

מבט 1 על תאגד:

המבט מאפשר לקבל תמונה מאוחדת של האימונים, כולל פרטי המפקדים והמחלקות המשתתפים בכל אימון. כאשר נשתמש במבט, לא נצטרך לבצע את כל החיבורים הללו בכל פעם מחדש, אלא נוכל לבצע שאילתות פשוטות על המבט כדי לקבל את כל המידע.

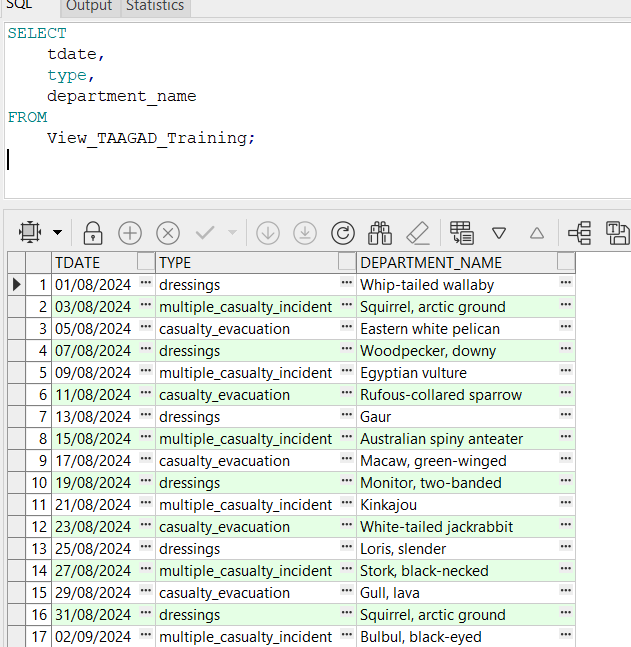
תמונה שמכילה טקסט, צילום מסך, תצוגה, גופן

התיאור נוצר באופן אוטומטי

שליפת הנתונים:  
  


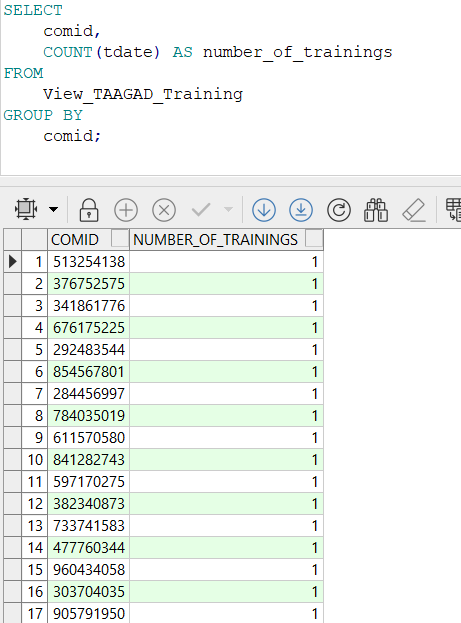
שאילתה 1:

השאילתה מחזירה רשימה של כל האימונים המתועדים יחד עם תאריך, סוגי האימונים ושמות המחלקות המשתתפות.



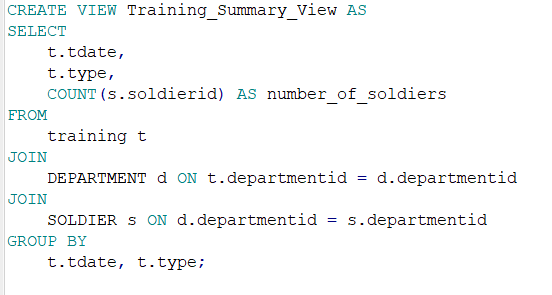
שאילתה 2:

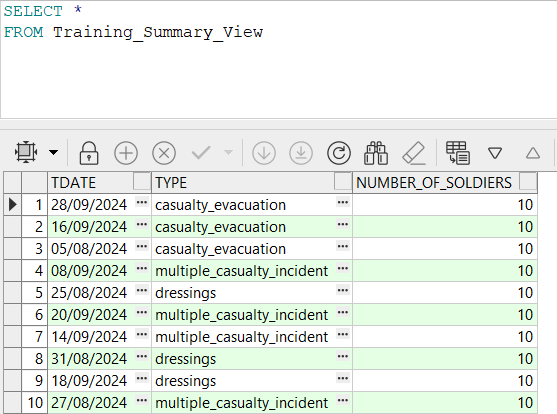
השאילתה מחזירה את מספר האימונים של כל מפקד.



מבט 2 על חיל ההנדסה:

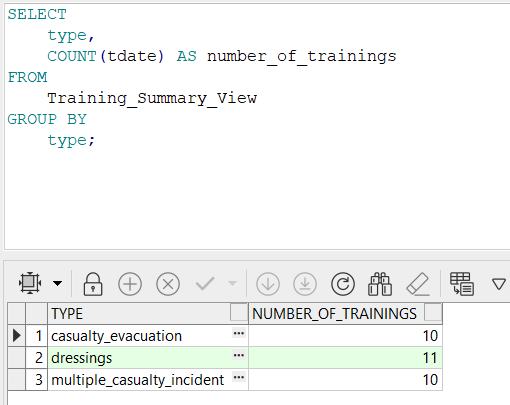
המבט מציג את הפרטים של כל אימון - תאריך, סוג ומספר החיילים שאומנו בכל אימון. כאשר נשתמש במבט, לא נצטרך לבצע את כל החיבורים הללו בכל פעם מחדש, אלא נוכל לבצע שאילתות פשוטות על המבט כדי לקבל את כל המידע.



שליפת הנתונים:  


שאילתה 1:

השאילתה תחזיר את סוגי האימונים ומספר הפעמים שכל סוג אימון התקיים.



שאילתה 2:

השאילתה מחזירה את תאריך האימון וסוג האימון רק עבור האימונים שבהם השתתפו יותר מ-8 חיילים.

